



DESCRIPCIÓN DE LA ASIGNATURA

Grado/Máster en:	Graduado/a en Ingeniería de Tecnologías de Telecomunicación por la Universidad de Málaga
Centro:	Escuela Técnica Superior de Ingeniería de Telecomunicación
Asignatura:	Programación 2
Código:	109
Tipo:	Formación básica
Materia:	FB-Informática
Módulo:	Formación Básica
Experimentalidad:	69 % teórica y 31 % práctica
Idioma en el que se imparte:	Español
Curso:	1
Semestre:	2
Nº Créditos	6
Nº Horas de dedicación del estudiante:	150
Nº Horas presenciales:	60
Tamaño del Grupo Grande:	72
Tamaño del Grupo Reducido:	30
Página web de la asignatura:	http://etsit.cv.uma.es/

EQUIPO DOCENTE

Departamento: LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN

Área: LENGUAJES Y SISTEMAS INFORMÁTICOS

Nombre y Apellidos	Mail	Teléfono Laboral	Despacho	Horario Tutorías
Coordinador/a: FRANCISCO JAVIER VEREDAS NAVARRO	franveredas@uma.es	952137155	3.2.42 - E.T.S. Ing. Telecomunicación	Primer cuatrimestre: Miércoles 10:00 - 14:00, Martes 10:00 - 12:00 Segundo cuatrimestre: Lunes 10:30 - 12:30, Viernes 12:30 - 14:30, Martes 12:30 - 14:30
NATALIA MORENO VERGARA	nmv@uma.es	+(34)952133393	3.2.34 - E.T.S.I. INFORMÁTICA	Todo el curso: Miércoles 17:30 - 19:00 Primer cuatrimestre: Lunes 11:00 - 14:00, Miércoles 09:30 - 11:00

RECOMENDACIONES Y ORIENTACIONES

* Necesidades formativas: para el seguimiento adecuado de la asignatura, es necesaria la comprensión y dominio de los conceptos de programación adquiridos en la asignatura de "Programación 1" impartida en el primer curso durante el primer semestre.

- Concepto de algoritmo y programa.
- Estructuras de control de flujo según el paradigma de programación estructurada.
- Análisis y construcción de programas utilizando una metodología de diseño descendente y refinamientos sucesivos (abstracción procedimental).
- Tipos de datos simples y compuestos: cadenas de caracteres, registros y arrays.
- Algoritmos de búsqueda y ordenación.
- Habilidades prácticas en el diseño, desarrollo y depuración de programas utilizando el lenguaje de programación C++ en sistemas operativos usuales (Linux y Windows).

* Necesidades transversales: capacidad de trabajo continuado y autosuficiencia en la organización del mismo, autosuficiencia en la adquisición y ampliación del conocimiento adquirido (manejo de bibliografías específicas, artículos técnicos y científicos, recursos electrónicos, etc.), habilidades de comunicación oral y escrita, nivel básico de inglés.

CONTEXTO

La asignatura se enmarca dentro del conjunto de asignaturas que proporcionan la formación básica de la titulación, y cuyos objetivos se centran en el desarrollo de habilidades de resolución de problemas y en la adquisición de conocimientos tecnológicos sobre el uso y programación de ordenadores. Así, esta asignatura se encuentra estrechamente relacionada con la asignatura de "Programación 1" del primer semestre de primer curso, y requiere de los conocimientos y habilidades adquiridos en ella. A su vez, proporciona un conjunto de conocimientos y habilidades necesarias para el adecuado seguimiento de la asignatura de "Software de Comunicaciones" del segundo semestre de segundo curso.

COMPETENCIAS

1 **Competencias generales y básicas. (Competencias generales de grados en RD 1393/2007)**

GENERA Todas la competencias generales de grados del RD 1393/2007: G01-G08.
LES_GR
ADO

2 **Competencias generales y básicas. (Competencias generales para Ingeniero Técnico de Telecomunicación en Orden CIN/352/2009)**



2 Competencias generales y básicas. (Competencias generales para Ingeniero Técnico de Telecomunicación en Orden CIN/352/2009)

- G-11** Conocimiento de materias básicas y tecnologías, que le capacite para el aprendizaje de nuevos métodos y tecnologías, así como que le dote de una gran versatilidad para
- G-12** Capacidad de resolver problemas con iniciativa, toma de decisiones, creatividad, y de comunicar y transmitir conocimientos, habilidades y destrezas, comprendiendo la responsabilidad ética y profesional de la actividad del Ingeniero Técnico de Telecomunicación.

3 Competencias específicas. (Formación básica establecida en Orden CIN/352/2009)

- FB-2** Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

CONTENIDOS DE LA ASIGNATURA

Programación II

Tema 1. Almacenamiento persistente de datos

- 1.1. Ficheros
- 1.2. Flujos de entrada y salida de datos

Tema 2. Tipos Abstractos de Datos

- 2.1. Programación modular
- 2.2. Clases, objetos, encapsulación
- 2.3. Introducción a la genericidad

Tema 3. Estructuras de datos dinámicas lineales

- 3.1. El tipo puntero
- 3.2. Gestión de memoria dinámica
- 3.3. Implementación de estructuras de datos dinámicas lineales

Tema 4. Programación Orientada a Objetos

- 4.1. Herencia
- 4.2. Polimorfismo y vinculación dinámica

Tema 5. Contenedores

- 5.1. Descripción y uso de los contenedores más usuales

ACTIVIDADES FORMATIVAS

Actividades presenciales

Actividades expositivas

Lección magistral

Actividades prácticas en instalaciones específicas

Prácticas en laboratorio

Actividades no presenciales

Actividades de discusión, debate, etc.

Discusiones

Actividades prácticas

Resolución de problemas
Desarrollo y evaluación de proyectos

Estudio personal

Estudio personal



ACTIVIDADES DE EVALUACIÓN

Actividades de evaluación no presenciales

Actividades de evaluación del estudiante

Otras actividades no presenciales eval.estudiante: Prácticas no presenciales: 4 h

Actividades de evaluación presenciales

Actividades de evaluación del estudiante

Examen parcial: Pruebas parciales: 4h

Examen final: Examen final: 4 h

Otras actividades eval.del estudiante: Pruebas presenciales de autoría: 3h

RESULTADOS DE APRENDIZAJE / CRITERIOS DE EVALUACIÓN

Con el fin de satisfacer las competencias indicadas y los objetivos del título, las actividades de esta materia estarán encaminadas a obtener los siguientes resultados de aprendizaje:

1. Comprender la filosofía de diseño orientado a objetos y los conceptos de encapsulación, abstracción, herencia y polimorfismo.
AE: Examen final y Otras (pruebas parciales, prácticas no presenciales y pruebas presenciales de autoría).
2. Conocer las principales características y funcionalidades de los sistemas de almacenamiento: ficheros y bases de datos.
AE: Examen final y Otras (pruebas parciales, prácticas no presenciales y pruebas presenciales de autoría).
3. Saber utilizar estructuras avanzadas de datos proporcionadas como colecciones en los lenguajes de programación orientados a objetos.
AE: Examen final y Otras (pruebas parciales, prácticas no presenciales y pruebas presenciales de autoría).
4. Conocer técnicas de gestión de memoria dinámica y estructuras de datos enlazadas lineales.
AE: Examen final y Otras (pruebas parciales, prácticas no presenciales y pruebas presenciales de autoría).

Los resultados de aprendizaje cubren las competencias específicas detalladas.

PROCEDIMIENTO DE EVALUACIÓN

La evaluación de la asignatura se basará fundamentalmente en la capacidad del alumno para diseñar e implementar sistemas software de complejidad media utilizando el lenguaje orientado a objetos C++, así como en la realización de prácticas de programación durante el curso lectivo.

* Actividades Presenciales:

AEP 1. Actividades de evaluación del estudiante

AEP 1.5. Examen Final. En la convocatoria 1ª ordinaria (junio), la evaluación se realizará a través de un examen final en el laboratorio, cuyo peso sobre la calificación final del alumno será del 70%. En la convocatoria 2ª ordinaria (septiembre) el peso del examen de laboratorio sobre la calificación final del alumno será la mejor opción de las dos siguientes: 70% (+ 30% de evaluación continua) ó 100% (+ 0% de evaluación continua). En las convocatorias extraordinarias, el peso del examen final (a realizar también en el laboratorio) será el 100% de la evaluación de la asignatura.

AEP 1.8. Otras: En las convocatorias ordinarias, para obtener la calificación correspondiente a las prácticas de programación evaluables no presenciales (AENP 1.4), los alumnos deberán realizar satisfactoriamente las pruebas presenciales de autoría requeridas por el profesor.

* Actividades No Presenciales:

AENP 1. Actividades de evaluación del estudiante

AENP 1.4. Otras: Prácticas no presenciales de programación. Se publicarán prácticas evaluables a realizar por el alumno, de forma no presencial, durante el semestre lectivo, que deberán ser entregadas para su evaluación en la primera convocatoria ordinaria, y no serán recuperables en la segunda convocatoria ordinaria, es decir, la calificación de las prácticas obtenida en la primera convocatoria ordinaria será también utilizada para la segunda convocatoria ordinaria. Dichas prácticas evaluables estarán acompañadas de pruebas presenciales (AEP 1.8) que tendrán como objeto verificar la autoría por parte del alumno. En la convocatoria 1ª ordinaria (junio), el valor total de las prácticas y pruebas asociadas será del 30% de la calificación final (+ 70% del examen). En la convocatoria 2ª ordinaria (septiembre), el valor total de las prácticas y pruebas asociadas será la mejor opción de las dos siguientes: 30% (+ 70% del examen) o 0% (+ 100% del examen). Nótese que en las convocatorias extraordinarias, la evaluación de las prácticas no será tenida en cuenta. El plagio, total o parcial, de las prácticas no presenciales y/o de alguna de sus pruebas de autoría asociadas, conllevará una valoración de 0 puntos en el total de las prácticas no presenciales (30% de la evaluación) en ambas convocatorias ordinarias.

ESTUDIANTES A TIEMPO PARCIAL Y DEPORTISTAS UNIVERSITARIOS DE ALTO NIVEL:

- Examen final: requiere asistencia en la fecha y hora oficial.

- Práctica no presencial evaluable de programación: solo la prueba de autoría asociada a la misma requiere asistencia en la fecha y hora previstas, la



cual podrá ser acordada con el profesor en el caso de que fuera imposible su asistencia, siempre con previo aviso y justificación.

BIBLIOGRAFÍA Y OTROS RECURSOS

Básica

Cómo programar en C/C++. Prentice-Hall, 1994.; H. M. Deitel, P. J. Deitel
Ejercicios de programación creativos y recreativos en C++. Prentice-Hall, 2002.; C. G. Rodríguez, L.F. Llana, R. Martínez, P. Palao, C. Pareja
El lenguaje de programación C++. Edición Especial, Addison-Wesley, 2001; Stroustup, B
Programación en C++. Algoritmos, estructuras de datos y objetos. Ed. McGrawHill, 2000.; Joyanes Aguilar, L.
Resolución de problemas con C++. Segunda Edición. , Prentice Hall 2000; Savitch, W.

Complementaria

Accelerated C++. Practical programming by example. A. Koenig and B.E. Moo. Addison-Wesley 2000.
Thinking in C++: introduction to standard C++. B. Eckel. Volume One, 2nd Edition. Prentice-Hall 2000.

DISTRIBUCIÓN DEL TRABAJO DEL ESTUDIANTE

ACTIVIDAD FORMATIVA PRESENCIAL

Descripción	Horas	Grupo grande	Grupos reducidos
Lección magistral	41,4	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Prácticas en laboratorio	18,6	<input type="checkbox"/>	<input checked="" type="checkbox"/>

TOTAL HORAS ACTIVIDAD FORMATIVA PRESENCIAL 60

ACTIVIDAD FORMATIVA NO PRESENCIAL

Descripción	Horas
Resolución de problemas	27,6
Desarrollo y evaluación de proyectos	27,6
Discusiones	6
Estudio personal	13,8

TOTAL HORAS ACTIVIDAD FORMATIVA NO PRESENCIAL 75

TOTAL HORAS ACTIVIDAD EVALUACIÓN 15

TOTAL HORAS DE TRABAJO DEL ESTUDIANTE 150

