

## PLAN DOCENTE DE LA ASIGNATURA

Curso académico: 2024/2025

Identificación y características de la asignatura			
Código	501383	Créditos ECTS	6
Denominación (español)	Fundamentos de programación		
Denominación (inglés)	Fundamentals of programming		
Titulaciones	Grado en Ingeniería de Sonido e Imagen en Telecomunicación		
Centro	Escuela Politécnica		
Semestre	1	Carácter	Obligatorio
Módulo	Formación básica		
Materia	Informática		
Profesorado			
Nombre	Despacho	Correo-e	Página web
Cristina Vicente Chicote	42	<a href="mailto:cristinav@unex.es">cristinav@unex.es</a>	<a href="https://bit.ly/3yH5tUh">https://bit.ly/3yH5tUh</a>
Área de conocimiento	Lenguajes y sistemas informáticos		
Departamento	Ingeniería de sistemas informáticos y telemáticos		
Profesor/a coordinadora (si hay más de uno)	Cristina Vicente Chicote		
Competencias			
Competencias básicas			
<b>CB1.</b>	Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.		
<b>CB2.</b>	Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.		

<b>CB3.</b>	Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.
<b>CB4.</b>	Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
<b>CB5.</b>	Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.
<b>Competencias específicas del módulo de formación básica</b>	
<b>CP2.</b>	Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
<b>CT1.</b>	Aplicar en su vida profesional las TIC y todos los desarrollos que vayan surgiendo de ellas, como la comunicación a través de Internet y, en general, manejo de herramientas multimedia para la comunicación a distancia.
<b>CT3.</b>	Redactar informes técnicos sobre soluciones a problemas asociados al campo de las Telecomunicaciones con el necesario rigor científico y tecnológico.
<b>CT9.</b>	Habilidades interpersonales asociadas a la capacidad de relación con otras personas y de trabajo en grupo. Habilidades para trabajar en equipos multidisciplinares con profesionales de áreas afines en empresas o instituciones públicas ligadas a la innovación tecnológica en el ámbito de las Telecomunicaciones. Habilidades para liderar grupos de trabajo en el campo de las Telecomunicaciones.
<b>Resultados de aprendizaje de la materia asociados, parcialmente, a esta asignatura:</b>	
<b>OG3.</b>	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 adaptarse a nuevas situaciones.
<b>OG4.</b>	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.
<b>OG6.</b>	Facilidad para el manejo de especificaciones, reglamentos y normas de obligado cumplimiento.
<b>OG9.</b>	Capacidad de trabajar en un grupo multidisciplinar y en un entorno multilingüe y de comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.

Contenidos
Breve descripción del contenido
Fundamentos de la programación. Tipos de datos y estructuras de control. Estructuras de datos básicas. Lenguajes estructurados para el desarrollo de sistemas software. Introducción al diseño e implementación de bases de datos y sistemas de información.
Temario de la asignatura
<p>Denominación del tema 1: <b>Introducción a las bases de datos</b></p> <p>Contenidos del tema 1:</p> <ol style="list-style-type: none"> <li>1.1. Importancia de las bases de datos.</li> <li>1.2. Introducción al diseño de bases de datos.</li> <li>1.3. Elementos básicos: tablas y relaciones.</li> <li>1.4. Diseño de consultas, formularios e informes.</li> </ol> <p>Descripción de las actividades prácticas del tema 1: sesiones de laboratorio sobre diseño y uso de bases datos.</p>
<p>Denominación del tema 2: <b>Introducción a la programación.</b></p> <p>Contenidos del tema 2:</p> <ol style="list-style-type: none"> <li>2.1. Conceptos básicos: algoritmo, proceso y procesador.</li> <li>2.2. Codificación y almacenamiento de la información.</li> <li>2.3. Entornos de programación.</li> <li>2.4. Primeros programas.               <ol style="list-style-type: none"> <li>2.4.1. Estructura de un programa.</li> <li>2.4.2. Variables y tipos de datos simples.</li> <li>2.4.3 Operaciones y expresiones.</li> </ol> </li> </ol> <p>Descripción de las actividades prácticas del tema 2: sesiones de laboratorio en la que los estudiantes desarrollarán sus primeros programas sencillos en C++.</p>
<p>Denominación del tema 3: <b>Programación modular</b></p> <p>Contenidos del tema 3:</p> <ol style="list-style-type: none"> <li>3.1. Principios de la programación modular. Diseño descendente.</li> <li>3.2. Módulos: concepto, tipos y parámetros.</li> <li>3.3. Especificación: pre- y post-condiciones.</li> <li>3.4. Bibliotecas.</li> </ol> <p>Descripción de las actividades prácticas del tema 3: sesiones de laboratorio sobre programación modular.</p>
<p>Denominación del tema 4: <b>Programación estructurada</b></p> <p>Contenidos del tema 4:</p> <ol style="list-style-type: none"> <li>4.1. Principios de la programación estructurada.</li> <li>4.2. Estructuras condicionales.</li> <li>4.3. Estructuras de repetición.</li> </ol> <p>Descripción de las actividades prácticas del tema 4: sesiones de laboratorio sobre programación estructurada.</p>

Denominación del tema 5: **Análisis de algoritmos**

Contenidos del tema 5:

- 5.1. Análisis de eficiencia: tiempo de ejecución y uso de memoria.
- 5.2. Órdenes de complejidad. Notación O.
- 5.3. Resolución de ejemplos y problemas.

Descripción de las actividades prácticas del tema 5: --

Denominación del tema 6: **Recursividad**

Contenidos del tema 6:

- 6.1. Conceptos básicos.
- 6.2. Clasificación.
- 6.2. Traza de ejecución.

Descripción de las actividades prácticas del tema 6: sesiones de laboratorio sobre recursividad.

Denominación del tema 7: **Registros, vectores y matrices**

Contenidos del tema 7:

- 7.1. Registros: definición y operaciones básicas.
- 7.2. Vectores:
  - 7.2.1. Definición y operaciones básicas.
  - 7.2.2. Algoritmos de búsqueda y ordenación.
- 7.3. Matrices. Representación de datos n-dimensionales.
- 7.4. Cadenas de caracteres.

Descripción de las actividades prácticas del tema 7: sesiones de laboratorio sobre el uso de registros, vectores y matrices.

Denominación del tema 8: **Definición y uso de nuevos tipos de datos**

Contenidos del tema 8:

- 8.1. Tipos abstractos de datos (TAD).
- 8.2. Especificación, implementación y uso.

Descripción de las actividades prácticas del tema 8: sesiones de laboratorio sobre la implementación y el uso de TADs.

Denominación del tema 9: **Ficheros**

Contenidos del tema 9:

- 9.1. Utilidad del uso de ficheros.
- 9.2. Tipos de ficheros: entrada, salida y entrada/salida.
- 9.3. Operaciones básicas: apertura, lectura/escritura y cierre.

Descripción de las actividades prácticas del tema 9: sesiones de laboratorio sobre el uso de ficheros.

Actividades formativas								
Horas de trabajo del alumno/a por tema		Horas Gran grupo	Actividades prácticas				Actividad de seguimiento	No presencial
Tema	Total		CH	L	O	S		
1	21,5	2		3			1,5	15
2	9,5	3		1,5			-	5
3	16,5	5		1,5			-	10
4	16,5	5		1,5			-	10
5	16	6		-			-	10
6	17,5	6		1,5			-	10
7	17,5	6		1,5			-	10
8	22	9		1,5			1,5	10
9	7,5	1		1,5			-	5
<b>Evaluación</b>	<b>5,5</b>	<b>2</b>		<b>1,5</b>				<b>2</b>
<b>TOTAL</b>	<b>150</b>	<b>45</b>		<b>15</b>			<b>3</b>	<b>87</b>

GG: Grupo Grande (85 estudiantes).

CH: Actividades de prácticas clínicas hospitalarias (7 estudiantes)

L: Actividades de laboratorio o prácticas de campo (15 estudiantes)

O: Actividades en sala de ordenadores o laboratorio de idiomas (20 estudiantes)

S: Actividades de seminario o de problemas en clase (40 estudiantes).

TP: Tutorías Programadas (seguimiento docente, tipo tutorías ECTS).

EP: Estudio personal, trabajos individuales o en grupo, y lectura de bibliografía.

### Metodologías docentes

Se combinan distintos tipos de actividades formativas orientadas a la adquisición, por parte de los estudiantes, de las competencias y resultados de aprendizaje cubiertos en la asignatura. Se utiliza un enfoque de "aprender haciendo" ("*learn by doing*") que pone al estudiante en el centro del proceso de enseñanza-aprendizaje, proponiéndole actividades que le permiten y animan a participar de forma activa en el mismo. En ocasiones, tanto en las clases de teoría como en las prácticas de laboratorio, se utiliza una metodología de clase invertida (*flipped classroom*), o se proponen retos (individuales, en pareja o en grupos más grandes) en los que los estudiantes deben aplicar lo aprendido. Se fomenta el uso de buenas prácticas de programación y se hace hincapié en la importancia que tiene diseñar y desarrollar software de calidad.

### Resultados de aprendizaje

El estudiantado es capaz de...

- Implementar una base de datos sencilla utilizando un sistema gestor de BB. DD.
- Explicar el funcionamiento de código escrito en un lenguaje de programación estructurado.
- Implementar algoritmos que manejen estructuras de datos básicas para resolver problemas simples.
- Seleccionar las estructuras de datos y los algoritmos básicos sobre ellas más adecuados para implementar programas eficientes.
- Utilizar las técnicas y herramientas básicas empleadas en el diseño y el desarrollo de programas.

- Desarrollar programas de tamaño medio, escritos en un lenguaje estructurado, aplicando todas las etapas de un método de construcción de software, individualmente y en grupo.

## Sistemas de evaluación

Para poder evaluar la consecución de los objetivos de aprendizaje de esta asignatura en relación con las competencias técnicas y transversales asignadas, se han considerado adecuados los siguientes instrumentos de evaluación:

- Portafolio
- Proyectos
- Pruebas escritas

Estos instrumentos generales de evaluación, que integran otros más simples y directos, permiten realizar un análisis completo de los niveles de consecución de las distintas competencias cubiertas en la asignatura.

### Portafolio

El portafolio estará formado por distintas actividades simples que se propondrán a los estudiantes a lo largo del semestre. Estas actividades podrán ser individuales o en grupo y podrán plantearse durante las clases presenciales (en grupo grande o en el laboratorio) o bien como parte de su trabajo fuera del aula. El portafolio podrá incluir actividades de distintos tipos: resolución de problemas, resolución de cuestionarios, coevaluación de los trabajos entregados por otros estudiantes, debates, etc.

Además del valor individual de cada actividad incluida en el portafolio como herramienta de evaluación, este instrumento permitirá, al estudiante, observar el trabajo realizado a lo largo del curso y reflexionar sobre su evolución y, al profesor, fomentar el trabajo continuo y ofrecer a tiempo al estudiante el *feedback* necesario para que pueda mejorar y alcanzar el nivel deseado al final del semestre.

### Proyectos

A lo largo del semestre se propondrán dos proyectos: uno de bases de datos y otro de programación. Estos proyectos se desarrollarán de manera individual o por parejas, según se indique en el correspondiente enunciado.

Los proyectos de programación son instrumentos de evaluación que permiten evaluar muchas de las competencias técnicas y transversales de la asignatura ya que, para su realización, los estudiantes deben demostrar ser capaces de: (1) analizar el problema que se les plantea; (2) diseñar una o más posibles soluciones; (3) comparar dichas soluciones, por ejemplo, en términos de eficiencia, modularidad, extensibilidad, etc.; (4) implementar, probar y documentar la mejor de ellas; y, eventualmente, si el profesorado de la asignatura lo considera necesario, (5) defender su implementación en un examen escrito o una entrevista personal.

### Pruebas escritas

Para asegurar la adquisición de los conocimientos mínimos desarrollados en la asignatura se realizarán pruebas escritas que podrán incluir: preguntas de tipo test, preguntas cortas, resolución de problemas, etc.

### Crterios de evaluación

Tal y como contempla en la normativa vigente de evaluación de la Universidad de Extremadura, esta asignatura puede superarse siguiendo las modalidades de evaluación

continua y de evaluación global. La elección de una u otra modalidad para cada convocatoria corresponde al estudiante durante el primer cuarto del semestre en el que se imparte la asignatura o hasta el último día del periodo de ampliación de matrícula, si éste fuera posterior (Art. 4.5 de la Normativa de Evaluación de la UEx, DOE 3 de noviembre de 2020).

### **Modalidad de evaluación continua:**

- Para superar esta asignatura en la modalidad de evaluación continua, los estudiantes deberán superar los requisitos mínimos asociados a los cuatro bloques de la asignatura: *portafolio*, *proyecto de bases de datos*, *proyecto de programación* y *pruebas escritas*, asociados a los instrumentos de evaluación previamente descritos, tal y como se detalla a continuación.
- La puntuación de cada bloque se calculará sobre 10.
- La nota de un bloque superado (cumplidos los requisitos mínimos) se guardará durante todas las convocatorias del curso, siempre que el estudiante tenga derecho a examen en la convocatoria en la que se supere el bloque.

### **Bloque 1: Portafolio**

- La nota de este bloque, **NPOR**, representa el 10% de la nota final.
- Este bloque se superará realizando una serie de actividades que se irán proponiendo a lo largo del curso. Según su naturaleza, estas actividades podrán realizarse en el aula (durante las clases de teoría o en los laboratorios de prácticas) o fuera del aula (en horario no presencial, a través del aula virtual).
- La asistencia a las clases de teoría, las sesiones de laboratorio y las tutorías programadas podrá considerarse, a efectos de la evaluación, como parte de las actividades del portafolio.
- La nota de este bloque se obtiene a partir de la ponderación de las calificaciones obtenidas en las actividades propuestas.
- No es necesario obtener una nota mínima en este bloque para considerarlo superado.
- La nota obtenida en este bloque no es recuperable en las convocatorias extraordinarias del curso.

### **Bloque 2: Proyecto de bases de datos**

- La nota de este bloque, **NPBD**, representa el 15% de la nota final.
- Para superar este bloque, deberá obtenerse una nota mínima de 5 sobre 10.
- Para superar el proyecto de bases de datos será imprescindible: (1) utilizar adecuadamente el sistema gestor de bases de datos propuesto; (2) entregar la base de datos y la documentación solicitada, cumpliendo todos los requisitos establecidos en el correspondiente enunciado; y (3) superar la defensa que se plantee.
- Cuando la nota obtenida en este proyecto sea superior a 5 sobre 10, éste se considerará superado y la nota se guardará durante todas las convocatorias del curso a las que el estudiante tenga derecho.
- Es responsabilidad de cada estudiante la custodia y protección de sus proyectos.

### **Bloque 3: Proyecto de programación**

- La nota de este bloque, **NPP**, representa el 30% de la nota final.
- Para superar este bloque, deberá obtenerse una nota mínima de 5 sobre 10.
- Para superar el proyecto de programación será imprescindible: (1) utilizar adecuadamente el entorno de programación propuesto; (2) entregar el programa y la documentación solicitada, cumpliendo todos los requisitos establecidos en el correspondiente enunciado; y (3) superar la defensa que se plantee.
- Cuando la nota obtenida en este proyecto sea superior a 5 sobre 10, éste se considerará superado y la nota se guardará durante todas las convocatorias del curso a las que el estudiante tenga derecho.
- Es responsabilidad de cada estudiante la custodia y protección de sus proyectos.

### **Bloque 3: Pruebas escritas**

- La nota de este bloque, **NPRU**, representa el 45% de la nota final.
- Para superar este bloque es necesario obtener una nota mínima de 5 sobre 10.
- Aquellos estudiantes que superen el examen parcial que se realizará durante el curso, sólo tendrán que examinarse en la convocatoria ordinaria de la segunda parte de la asignatura (contenidos no evaluados en el parcial). Aquellos que no se presenten o no superen el examen parcial, deberán superar un examen global en el que se evaluarán todos los contenidos de la asignatura.
- Quienes no superen este bloque en la convocatoria ordinaria, podrán recuperarlo en cualquiera de las convocatorias extraordinarias a las que tengan derecho, realizando una única prueba global en la que se evaluarán todos los contenidos de la asignatura, independientemente de si aprobaron o no el examen parcial. Esto es: la nota del examen del parcial no se guardará para las convocatorias extraordinarias.
- Cada prueba, parcial o final, podrá incluir preguntas de tipo test, preguntas cortas, resolución de problemas, etc. Podrán establecerse requisitos adicionales sobre la nota mínima que debe obtenerse en cada parte del examen (por ejemplo, podrá requerirse una nota mínima en los problemas para considerar superada la prueba).

### **Modalidad de evaluación global:**

- Para superar esta asignatura por evaluación global, los estudiantes deberán superar los requisitos mínimos asociados a los tres bloques siguientes de la asignatura: *proyecto de bases de datos*, *proyecto de programación* y *pruebas escritas*, tal y como se detalla a continuación.
- La puntuación de cada bloque se calculará sobre 10.
- La nota de una parte superada no se guardará para ninguna convocatoria posterior del curso.

### **Bloque 1: Proyecto de bases de datos**

- La nota de este bloque, **NPBD**, representa el 15% de la nota final.
- Para superar este bloque, deberá obtenerse una nota mínima de 5 sobre 10.

- Para superar el proyecto de bases de datos será imprescindible: (1) utilizar adecuadamente el sistema gestor de bases de datos propuesto; (2) entregar la base de datos y la documentación solicitada, cumpliendo todos los requisitos establecidos en el correspondiente enunciado; y (3) superar la defensa que se plantee.
- Es responsabilidad de cada estudiante la custodia y protección de sus proyectos.

### **Bloque 3: Proyecto de programación**

- La nota de este bloque, **NPP**, representa el 35% de la nota final.
- Para superar este bloque, deberá obtenerse una nota mínima de 5 sobre 10.
- Para superar el proyecto de programación será imprescindible: (1) utilizar adecuadamente el entorno de programación propuesto; (2) entregar el programa y la documentación solicitada, cumpliendo todos los requisitos establecidos en el correspondiente enunciado; y (3) superar la defensa que se plantee.
- Es responsabilidad de cada estudiante la custodia y protección de sus proyectos.

### **Bloque 3: Pruebas escritas**

- La nota de este bloque, **NPRU**, representa el 50% de la nota final de la asignatura.
- Para superar este bloque es necesario obtener una nota mínima de 5 sobre 10.
- En todas las convocatorias (ordinaria y extraordinarias), el examen evaluará todos los contenidos de la asignatura, pudiendo incluir preguntas de tipo test, preguntas cortas, resolución de problemas, etc. Podrán establecerse requisitos adicionales sobre la nota mínima que debe obtenerse en cada parte del examen (por ejemplo, podrá requerirse una nota mínima en los problemas para considerar superada la prueba).

### **Cálculo de la nota final de la asignatura**

- La copia o el plagio en cualquier actividad, proyecto o prueba, ya sea en una parte o en su totalidad, supone una nota final de SUSPENSO (0) en la convocatoria para todas las personas implicadas, además de las actuaciones legales pertinentes.
- En la modalidad de **evaluación continua**, si se cumplen todos los requisitos mínimos de los cuatro bloques considerados, la nota final se calculará como sigue:

$$\text{Nota final} = 0,1 * \text{NPOR} + 0,15 * \text{NPBD} + 0,3 * \text{NPP} + 0,45 * \text{NPRU}$$

- En la modalidad de **evaluación global**, si se cumplen todos los requisitos mínimos de los tres bloques considerados, la nota final se calculará como sigue:

$$\text{Nota final} = 0,15 * \text{NPBD} + 0,35 * \text{NPP} + 0,5 * \text{NPRU}$$

- Si no se cumplen todos los requisitos mínimos, la nota final será 2 o 3, dependiendo de los casos. La siguiente tabla muestra cómo se calculará la nota final, según los resultados obtenidos en los bloques de proyectos y pruebas escritas:

Proyectos	NP	NCR	Nota	NP	NCR/Nota	NP/NCR	Nota
Pruebas escritas	NP	NP	NP	NCR	NCR	Nota	Nota
<b>Nota final (*)</b>	<b>NP</b>	<b>2</b>	<b>3</b>	<b>2</b>	<b>3</b>	<b>3</b>	<b>Nota</b>

NP: no presentado a ese bloque

NCR: no cumple los requisitos mínimos de ese bloque

Nota: nota obtenida en el bloque, superados los requisitos mínimos.

(\*) Obtendrá una calificación de "NP" en la convocatoria ordinaria quien no haya superado ningún bloque por evaluación continua y no se presente a ninguna prueba durante el periodo de exámenes de esa convocatoria. En cada una de las convocatorias extraordinarias, obtendrá una calificación de "NP" quien no se presente a ninguna de las pruebas realizadas en dicha convocatoria.

### Bibliografía (básica y complementaria)

#### Bibliografía básica

- Walter Savitch, Resolución de problemas con C++. 5ª edición. Pearson-Addison Wesley, 2006
- Fatos Xhafa y otros, Programación en C++ para ingenieros. Thomson, 2006
- Jesús J. García Molina y otros. Una introducción a la programación. Un enfoque algorítmico. Thomson-Paraninfo, 2005

#### Bibliografía complementaria

- H.M. Deitel y P.J. Deitel, C++. Cómo programar. 6ª edición. Pearson - Prentice Hall, 2008.
- Luis Joyanes, Fundamentos de programación. Algoritmos, estructuras de datos y objetos. 4ª edición. McGraw-Hill, 2008.
- Luis Joyanes, Fundamentos de programación. Libro de problemas. 2ª edición. McGraw-Hill, 2002.
- J. Castro y otros, Curso de programación. McGraw-Hill, 1993.
- Luis Joyanes, Programación en C++. Algoritmos, estructuras de datos y objetos. McGraw-Hill, 2000.

### Otros recursos y materiales docentes complementarios

#### Medios materiales utilizados

- Pizarra
- Cañón de vídeo
- Ordenador

**Materiales y recursos utilizados.** Todos los materiales y recursos de la asignatura están disponibles en versión electrónica a través del aula virtual:

- Ficha 12a de la asignatura.
- Agenda del curso.

- Diverso material docente (presentaciones, vídeos, apuntes, etc.) para cada tema.
- Guiones de las sesiones de laboratorio.
- Proyectos de programación (completos y de trabajo).
- Listas de problemas (propuestos y resueltos).
- Código fuente en C++.
- Guías de buenas prácticas.

En la asignatura se utilizarán, entre otros, los siguientes recursos del aula virtual:

- Sistemas de participación: foros de comunicación; tablón de anuncios y novedades.
- Consultas: elección de la modalidad de evaluación; elección de grupo de prácticas.
- Tareas: entrega de trabajos; cuestionarios autoevaluables; actividades VPL (*Virtual Programming Lab*), etc.
- Evaluación y registro de actividad: libro de calificaciones; registro de asistencia a los laboratorios de prácticas.
- Información adicional: enlaces a páginas relacionadas con la asignatura.